

**IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE**

TITLE:

System and Method for Analyzing Computer Intelligible Electronic Data.

INVENTOR(S):

Steven E. Adams

Steven R. Zimmerman

James R. Davee

Fouzia S. Kiani

ASSIGNEE:

EC Outlook, Inc.

1 [0001] This patent application claims priority from a provisional patent application entitled "System
2 and Method for Analyzing and Describing Electronic Data, and generating Major and Minor Variant
3 Samples of Electronic Data," Serial No. 60/314,715, having a filing date of 8-24-01. This patent
4 application is also a continuation in part of another utility patent application entitled "System and
5 Method for Conducting Electronic Commerce," Serial No. 09/767,442 having a filing date of 1-19-
6 01.

7 **FIELD OF THE INVENTION**

8 [0002] The present invention relates generally to a system and method of analyzing electronic data
9 and, more particularly, to a system and method of determining the inherent structure of one or more
10 incoming data files and generating output data for use in retrieving or testing electronic data.

11

12 **BACKGROUND OF THE INVENTION**

13 [0003] The need for the efficient analysis of electronic data has become increasingly important as
14 reliance upon computer systems has increased. Electronic data, regardless of its type, benefits from
15 descriptive information capable of identifying and characterizing the individual data elements of
16 incoming data files. To illustrate, information describing the starting position, length, delimiting
17 character, etc., of individual data elements allows a database management system or other system to
18 more efficiently read and utilize an incoming data file.

19

20 [0004] The analysis of electronic data files requires descriptive information, whether found within
21 the data file or sourced externally, to identify and describe each data element. For example,
22 descriptive information allows database management systems to more efficiently extract data, extract
23 specific subsets of data, convert identified data into other formats, import data from other systems

1 and/or prepare external systems to utilize the incoming data file. If descriptive information is not
2 available, efficient use of the incoming data file is extremely difficult.

3

4 [0005] Typically, known systems utilize defined data formats to provide descriptive information for
5 electronic data files. Defined data formats, such as xBase, Excel, EDI or XML, contain descriptive
6 information which may be used to identify individual data elements of an incoming data file. Data
7 files having a defined format are typically referred to as structured files.

8

9 [0006] Although equipped with a predefined format, structured files such as EDI and XML are often
10 equipped with one or more implementation guides. These implementation guides provide additional
11 descriptive information for each element of the structured data file.

12

13 [0007] Some electronic data is produced without the benefit of a predefined format. This type of
14 electronic data is referred to as semi-structured data and is typically organized in a manner such that
15 individual data elements may be identified through data analysis. Specifically, the position of
16 individual data elements or the presence of delimiting characters within the semi-structured file may
17 be used to identify and describe the structural characteristics of each individual data element. The
18 organization of the data in this manner typically requires a painstaking process by the owner of the
19 data during extraction. Unfortunately, the process of organizing the data held within each data file is
20 time consuming and expensive.

21

22

23

SUMMARY OF THE INVENTION

2 [0008] Accordingly, the present invention provides a system and method of analyzing electronic
3 data that eliminates the need for externally sourced descriptions, thus reducing the time and expense
4 associated with manual creation of data file descriptive information. The present invention is
5 capable of automatically analyzing one or more incoming data files, generating information
6 descriptive of the structure of each data file and producing output data similar or identical in
7 structure to the incoming data file(s) for use in subsequent applications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 is a component diagram of one embodiment of the present invention.

[0010] Figure 2 is a flowchart of the data analysis process for structured data files of one embodiment of the present invention.

[0011] Figure 2A is a flowchart illustrating a portion of the record break analysis process of one embodiment of the present invention.

18 [0012] Figure 2B is a flowchart illustrating a portion of the field break analysis process of one
19 embodiment of the present invention.

21 [0013] Figure 3 is a flowchart of the data analysis process for semi-structured data files of one
22 embodiment of the present invention.

1 [0014] Figure 4 is an illustration of structured data file hierarchical representations of one
2 embodiment of the present invention.

3

4 [0015] Figure 5 is an illustration of semi-structured data file hierarchical representations of one
5 embodiment of the present invention.

6

7 [0016] Figure 6 is a flowchart illustrating a portion of the output generation process of one
8 embodiment of the present invention.

9

10 DETAILED DESCRIPTION OF THE INVENTION

11 [0017] The present invention is herein described as a computer implemented method of analyzing
12 electronic data, as a computer readable medium comprising a plurality of instructions for analyzing
13 computer intelligible electronic data and as a computer system for analyzing electronic data.
14 Referring to the Figures, the present invention is capable of analyzing electronic data to determine
15 the structural characteristics of the data. The structural characteristics may then be used to generate
16 output data comprising a structural map of the incoming data for use in a variety of applications.

17

18 [0018] Referring to Fig. 1, the present invention is equipped with a processing unit (12) capable of
19 reading and analyzing computer intelligible electronic data, as illustrated by Box (13). In one
20 embodiment, the present invention provides a storage device (14) electrically coupled to the
21 processing unit (12). In another embodiment, the present invention provides a user interface (15)
22 through which the user may view and/or modify output data (16). In one embodiment of the present
23 invention, only references to the source of output data (16) are stored within the storage device (14).

1 Specifically, the analyzed data files (20) themselves need not be stored, as illustrated by Box (11) of
2 Figure 1.

3

4 [0019] The present invention is highly versatile and may be used with a variety of hardware
5 platforms. For example, the present invention may be used with a host of personal computers and
6 mid-range computer platforms (not shown). Platform specific code may be generated for Windows,
7 Solaris, Linux, and Hewlett Packard HP-UX operating systems, if desired.

8

9 [0020] Any media type or environment supported by the operating system and hardware platform,
10 whether local to the system or over a network, may be used by the present invention. For example,
11 direct access storage devices (DASD), write-once read-many devices (WORM), directly accessible
12 tape and solid state devices (SSD), single or multiple read/write head, redundant array (RAID of any
13 level), or jukebox subsystems may be utilized by the present invention. The present invention is
14 capable of efficient operation without the use of proprietary media formats, hidden partitions, or any
15 other storage media preparation in addition to that required and/or supported by the operating system
16 and hardware platform on which the present invention is installed.

17

18 [0021] The present invention is capable of efficiently analyzing incoming data (20) regardless of its
19 type or structure. In one embodiment of the present invention, the term “data” is used to describe
20 actual characters or values such as a name (e.g. John Smith) or date (e.g. 5/7/01) stored in a
21 computer intelligible format. In another embodiment, data is accumulated into files (20). Files (20)
22 may take the form of a computer data file, a computer application, or any data input stream or data
23 collection introduced from an outside application or system. In one embodiment, these files (20)

1 may be divided into records (22) comprising a physical or logical division of the file (20) into one or
2 more sets of characters. In another embodiment, individual data elements retaining some
3 characteristic or value in addition to their simple character contents are referred to as a field (24).
4 For example, “2001” could be classified by value, year, and/or street number fields, depending upon
5 its intended use.

6

7 [0022] In one embodiment, the structure of the incoming data (20) is determined by analyzing the
8 syntactic and semantic characteristics of the incoming data. In one embodiment, syntax refers to the
9 physical characteristics of the fields (24) and/or records (22) present within the incoming data files.

10 For example, if a given field (24) contains the data “2001”, syntax would include a length of four
11 characters of the numeric type. Syntax may also include the field’s position within the record as
12 compared to other fields as well as the number of fields (24) in a record (22), the accumulated
13 character lengths of each field and record’s position within an electronic data file (20) as compared
14 to other records. Additionally, syntax may include the overall file size, the creation date, the last
15 modified date and the number of records (22) in the file (20). Syntax may also be used as a
16 validation test for data, as discussed below.

17

18 [0023] In one embodiment, semantics refers to the attribute characteristic values of a field (24),
19 record (22) or file (20). For example, if a given field (24) contains the data “2001”, semantics may
20 take the form of a “year” definition to describe the data and may also include definitions such as
21 “Ordered Items” or “Shipping Information”, depending on the type of data at issue. For any given
22 data file (20), semantics may include broad definitions such as “Company X Purchase Order” or

1 "XML Transaction Database". Semantic information is typically provided by the user upon creation
2 of the field (24), record (22) or file (20) at issue.

3

4 Data Analysis

5 [0024] The present invention is capable of receiving and analyzing one or more incoming data files
6 (20) to produce output data (16) capable of providing a concise description of the structural
7 characteristics of the incoming data (20). Data files to be analyzed are collected through a reading
8 process, as illustrated by Box (13) of Figure 1. In one embodiment, incoming data files (20) are not
9 imported or otherwise modified from their original state but are simply read by the processing unit
10 (12) of the present invention for analysis.

11

12

13

14

15

16

17

18

19

20

21

22

[0025] Although the present invention is capable of reading and analyzing individual electronic data files (20), it may be advantageous to combine substantially similar data files (18) for simultaneous analysis. Data files may be substantially similar in content and/or structure in that the files have at least one common characteristic. For example, electronic purchase orders and electronic invoices used by Company X, although distinct types of data files, may contain commonality. By analyzing similar files (18) simultaneously or as a continuous stream of data, the processing unit (12) of the present invention is capable of determining the structural characteristics of the electronic data files with greater accuracy. In short, a large number of files (18 and 20) having some degree of commonality will provide the system with additional examples of those possible structural configurations for the files, thus refining the analysis process.

1 [0026] Once similar data files (18 and 20), if any, have been grouped for reading the processing unit
2 (12) of the present invention is designed to automatically identify each electronic data field and its
3 associated structural data.

4

5 [0027] The system, upon reading electronic data, identifies the file type associated with the
6 incoming electronic data file(s) (20) as illustrated by Box (26) of Figures 2 and 3. In one
7 embodiment, the present invention identifies the incoming data file (20) as having a structured, semi-
8 structured or unstructured file type. In one embodiment, structured data refers to XML, EDI or other
9 “tagged” data formats. In another embodiment, semi-structured data refers to ASCII, flat, positional
10 or delimited file formats.

11

12 [0028] Referring to Figure 2, if the incoming data file has an explicitly named structure, the named
13 structure is used in conjunction with the incoming file (20) to break the file into records (22) and
14 fields (24). Specifically, the processing unit (12) of the present invention determines which
15 structure type is associated with the incoming data, as illustrated by Box (28) of Figure 2. In one
16 embodiment, the present invention maintains a library (not shown) to assist in identifying both the
17 file type (26) and the structure type (28) of the incoming data file (20). To illustrate, an incoming
18 XML file would first be identified by the present invention as having a structured file type (26). The
19 present invention may then access the library to determine which structure type (28) is exhibited by
20 the incoming file (20). In the present example, the processing unit (12) would determine that an
21 XML structure type is utilized using information describing the attributes of XML files stored within
22 the library.

23

1 [0029] Structured data sources provide syntactical and semantic information regarding the incoming
2 file (20) through the inherent structure representation of the file. Accordingly, a high percentage of
3 the syntactical and semantic information for structured incoming data files is automatically captured
4 and utilized by the present invention upon being read into the system (10).

5

6 [0030] Referring to Figs. 2 and 2A, once the file type (26) and structural type (28) have been
7 determined, the processing unit (12) of the present invention analyzes the electronic data file (20) to
8 identify the file's record break information, as illustrated by Box (30) of Figure 2. In one
9 embodiment, the present invention analyzes the structured file to identify record break characters
10 (32) typically used with the pre-determined file type. In one embodiment, record break information
11 (30) comprises demarcated record break characters (32) and/or character counts.

12

13 [0031] Utilizing the record break information (30) inherent within a structured file (20), the
14 processing unit (12) parses the electronic data file (20) into one or more electronic data records (22).
15 Data records (22) having substantially similar attributes are identified and matched by the processing
16 unit (12), as illustrated by Box (34) of Figure 2. In one embodiment, data records (22) are matched
17 by comparing syntactic information residing within the data file (20). For example, character counts
18 and common headings found within the incoming data file (20) may be used to denote substantially
19 similar records (22).

20

21 [0032] Referring to Figs. 2 and 2B, once individual records (22) have been identified, the processing
22 unit (12) of the present invention analyzes each individual record (22) in order to identify field break
23 information, as illustrated by Box (36) of Figure 2. In one embodiment, field break information

1 comprises demarcated field break characters (38) and/or transition points. The field break
2 information (38) is utilized to parse data records (22) into individual data fields (24). The processing
3 unit (12) of the present invention compares each individual data field (24) contained within
4 previously matched records (22) to establish syntactic values for the entire data file (20), as
5 illustrated by Box (62) of Figure 2. The data analysis process may be repeated as many times as is
6 necessary to determine each individual data record (22), field (24) and element, within the incoming
7 data file (20), as illustrated by Box (60) of Figure 2.

8

9 [0033] Referring to Figs. 2A and 3, the present invention is capable of determining the structural
10 characteristics of an incoming file (20) that has no explicitly named structure. To accomplish this,
11 the present invention first determines the file type (26) at issue. Data files (20) having no explicitly
12 named structure are read into the system for data analysis. The processing unit (12) of the present
13 invention analyzes the incoming data file(s) (20) to identify record break information, as illustrated
14 by Box (40) of Figure 3. In one embodiment, record break information comprises one or more line
15 termination characters (42) and/or character counts found within the data file (20). The record break
16 information (40) is utilized to parse the electronic data file (20) into one or more electronic data
17 records (22), as illustrated by Box (44). Data records (22) having substantially similar attributes are
18 identified and matched by the processing unit (12) of the present invention. In one embodiment,
19 data records (22) are matched by comparing syntactic information within the data file (20). For
20 example, character counts and common headings found within the incoming data file may be used to
21 denote substantially similar records, as illustrated by Box (44).

22

1 [0034] Referring to Figs. 2B and 3, once individual records (22) have been identified, the processing
2 unit (12) of the present invention analyzes each individual record (22) in order to identify field break
3 information (46). In one embodiment, field break information (46) comprises character type
4 transitions and/or character counts, as illustrated by Box (48) of Figure 3. The field break
5 information is utilized to parse data records into individual data fields. The processing unit (12) of
6 the present invention compares each individual data field contained within previously matched
7 records to ensure commonality between data fields, as illustrated by Box (50) of Figure 3. As with
8 data analysis of structural data files, the data analysis process for semi-structured or unstructured
9 files may be repeated as many times as is necessary to determine each individual data record (22),
10 field (24) and element (23) within the incoming data file (20), as illustrated by Box (70).

11
12 [0035] Referring to Figs. 2 and 3, once data analysis is completed, records and fields are identified
13 across all incoming data, as illustrated by Box (54). The structural patterns found during data
14 analysis may then be used to create the output data (16) of the present invention. In one
15 embodiment, the output data (16) created by the present invention contains a structural description of
16 at least a portion of the analyzed data file(s) (20). In one embodiment, the present invention creates
17 output data (16) describing the structural characteristics of the analyzed data file (20) or files as a
18 whole.

19
20 [0036] Each output (16), as created by the present invention, provides a concise description of the
21 analyzed data file(s) (20) providing the user with information about the analysis. This information
22 includes the identification of record types present within the incoming file(s), the structure of each
23 record, the sequence of record types, the cardinality of each record type, how records are grouped

1 together and whether the records are optional, required, vary in count, or repeat according to a
2 discernible pattern. In one embodiment, output data (16) is converted into a pre-selected computer
3 intelligible language (i.e., XML) that may then be stored within the storage device (14) for later use.

4

5 [0037] In one embodiment, the present invention uses tokenized symbology to denote the structure
6 of the analyzed data (20) as represented by the output data (16) of the present invention. The present
7 invention analyzes the inherent structure of the incoming data (20). In one embodiment, once the
8 structure of an incoming data file (20) has been determined, each analyzed data file is tokenized such
9 that each unique record (22) and field (24) is defined. Once tokenized, the structural characteristics
10 of the analyzed data are used to assign a symbolic identifier to each structural component of the data
11 file (20). Thus, the analyzed data (20) file represented by the output data (16) of the present
12 invention is assigned one or more tokenized symbols capable of symbolically representing the
13 structural characteristics (16) of the analyzed data.

14

15

Data Organization

16 [0038] Referring to Figures 4 and 5, the present invention is capable of describing the relationship
17 amongst and between each data element (23) within incoming data files (20). Specifically, the
18 present invention is capable of generating a hierarchical representation (52) of each record (22) and
19 field (24) within an analyzed data file (20). In one embodiment, each data element (23) is referred to
20 as a node and each node is a direct descendent of a parent node. By identifying the parentage of each
21 node, the present invention provides the user with a reference with which to determine and/or locate
22 the source file from which each data element belongs, as illustrated by Box (11) of Figure 1. In this
23 manner, each data element (23) has a defined position within the hierarchy as well as a defined

1 parentage, a specific organizational scheme and information from which the user may determine the
2 data element's siblings, collections and/or children. For the purpose of illustration only, the
3 following example of one of the naming conventions that may be used by the present invention is
4 provided as follows.

5

6 [0039] In one embodiment, a multi-level decimal point notation system may be used to identify each
7 file (20), record (22) and field (24). The decimal point notation system may also be valuable in
8 providing unique identification values for each data element (23). For example, if the value "1" is
9 used to describe a file (20) and all of its content, the first record of this file may be labeled "1.1"
10 while the second record may be labeled "1.2". Accordingly, the first field contained in the first
11 record "1.1" would be labeled as "1.1.1" while the second field in the first record labeled as "1.2.1".
12

13 [0040] In some cases, the data analysis process may indicate that a particular field (24) or record's
14 presence is optional and/or repeating. For example, an optional field or record may be designed
15 to carry an additional alpha character notation that is needed only in special cases. For example,
16 given two fields with node values of "1.2.3A" and "1.2.3B", the user and/or the processing unit (12)
17 of the present invention may quickly determine that field "3" in record "2" of file "1" may have two
18 distinct, yet valid, entries.

19

20 [0041] In one embodiment, the present invention utilizes two methodologies to describe the
21 hierarchy of nodes. The first methodology employed by one embodiment of the present invention
22 utilizes a structural description. To illustrate, an incoming file (20) may have several repeating
23 records (22) or fields (24) that are not useful in describing the structural characteristics of the file. In

1 this example, the structural node description is designed to limit the scope of the data analysis
2 process to the basic structure of the file (20), thus excluding the replication patterns of the above
3 mentioned records (22) and/or fields (24). In this example, the practical result is that the structural
4 description expresses only a single record instead of a plurality of repeating records or fields that do
5 not provide the system with structural information.

6

7 [0042] A second methodology employed by one embodiment of the present invention utilizes a data
8 sample node description. In one embodiment, the data sample node description displays each and
9 every data element (23) within the selected file (20), without regard to repetition or redundancy.
10 Specifically, this second methodology uses additional values in the node description to indicate the
11 iteration number of any repeating records (22) or fields (24). For example, node value of “(1-
12 1)1.2.1” and “(1-2)1.2.1” would represent first and second iterations of a repeating field with the
13 node value of “1.2.1”. In this example, “1.2.1” serves as a link between the simple structural
14 description and the data sample description.

15

16 Modification

17 [0043] Referring to Figs. 1, 4 and 5, a user interface (15) is provided by the present invention to
18 allow the display of both the hierarchical representation (52) of the structural characteristics of the
19 analyzed data file (20) and the specific semantic and syntactical information for each file, record
20 (22) and field (24). In one embodiment, the hierarchical representation (52) is expressed as an
21 expansion “tree” wherein the “root” denotes the file, the “branches” denote the records and the
22 “leaves” denote the fields. The expansion tree is suited to express the nodes and may take advantage

1 of the node methodologies described above. In one embodiment, node values describe the
2 root/branch/leaf construct through the multi-level decimal point methodology.

3

4 [0044] In another embodiment, a detailed collection of tables is used to list the structural
5 characteristics that may be reviewed and/or edited by the user. These values include, but are not
6 limited to:

7 - Minimum / Maximum Length

8 - Minimum / Maximum Value

9 - Justification

10 - Format

11 - Modified (True/False)

12 - Mandatory (True/False)

13 - Type (Alpha, Numeric, etc.)

14 - User-Defined Name

15

16 [0045] The present invention allows the user to view the structural characteristics gleaned from the
17 data analysis process and then modify same to achieve the proper results. Changes can be made to
18 individual records (22) or fields (24) by the user, as desired.

19

20 [0046] In one embodiment, the user interface (15) of the present invention is designed to limit the
21 displayed fields (24) to only those of the same type. For example, in a semi-structured file of
22 addresses, one field typically denotes "CITY". The display of this semi-structured file may be
23 filtered to only the "CITY" data field, thus allowing the user to review the structural characteristics

1 for that field (24). This feature of the present invention helps the user ensure that the automated data
2 analysis process accurately identifies the “CITY” field elements across all records. In another
3 embodiment, the list of “CITY” field elements is sorted alphabetically to allow for faster searching
4 and review by the user.

5

6 Output

7 [0047] Referring to Figs. 1, 2, 3 and 6, the present invention is capable of providing three types of
8 output data. The first type of output data (16TD) employed in one embodiment of the present
9 invention being output data designed for use by other applications intending to use data files (20)
10 analyzed by the present invention for their own input. The second type of output data (16T)
11 employed in one embodiment of the present invention being designed for use in generating different
12 versions of one or more incoming data files (20). The third type of output data (16) employed in one
13 embodiment of the present invention takes the form of generated data that describes the entire file
14 and the data elements and structure.

15

16 [0048] In one embodiment, the first type of output data (16TD) of the present invention may be
17 converted to an XML document as described above. The XML document is then used to parse and
18 translate documents being fed into another system, such as an electronic commerce system.
19 Structural characteristics are expressed within the XML document using normalized XML values
20 and expressions to enable them to be read and utilized by any system capable of reading and
21 processing an XML document. In short, the data analysis process of the present invention is used as
22 a parse command generator, thus enabling a subsequent user to describe an incoming data file (20) to
23 an external system.

1
2 [0049] Referring to Figure 6, in one embodiment, the second type of output data (16T) is a user
3 managed collection of output data (16) capable of matching the original incoming data file (20) with
4 the exception of one or more specific data fields (24A) containing user defined modifications (80).
5 This collection of output data may then be used to evaluate how the system reacts to these minor
6 changes. The present invention is data-centric in that it introduces no outside influences or
7 presumptions to the generations of the collection of output data (16T). Specifically, all output data
8 (16T) produced by the processing unit (12) of the present invention is sourced from the analyzed
9 data files (20) and varies only according to specific modification information (80) as supplied by the
10 user.

11
12 [0050] The present invention is capable of using the structural characteristics of analyzed files (20)
13 to create a plurality of output data (16T) identical to the analyzed data file (20) with the exception of
14 modified fields. Through the user interface (15), the user is given the opportunity to select specific
15 values (54) that will be used within each individual field (24) within the output data. In one
16 embodiment, the resulting output data (16T) utilize the original values of the incoming data file(s)
17 for all records (22) and fields (24) with the exception of a predetermined data field (24A). The value
18 (54) of this predetermined field (24A) is entered by the user as a modification instruction (80). In
19 one embodiment, if the modified field (24A) has more than one value (54) entered by the user, the
20 next output, or set of output, will use the second value (54B) from the user's entry, then the third
21 (54C), etc., until all of the user's values (54) have been used to produce output data (16).

22

1 [0051] Modification information (80) may take virtually any type (i.e., alpha, numeric, time, date,
2 etc.) or format as desired by the user. The present invention allows for the generation of output data
3 (16) that can purposefully fail, and through this failure, invoke additional error handling
4 programming. Accordingly, the present invention does not attempt to evaluate the impact (likely
5 success or failure of subsequent processing) of the user's modification. For example, it is possible
6 for the user to place characters into a numeric field for the purpose of causing expected error when
7 the output data(s) is parsed during subsequent processing. In one embodiment, the modification of
8 one or more fields (24) by the user, accumulated across substantially all fields in an analyzed data
9 file (20), may be used to generate output data (16) differentiated according to the modified field
10 (24A) only. To provide the user with convenient entry of modification information (80), one
11 embodiment of the present invention provides Field Increment Value Setting (FIVS).

12
13 [0052] The FIVS process of the present invention allows the user to manage numerical range values
14 which describe some collection of values for user modifications. For example, the user value "1-5"
15 is equal to entering user values of "1, 2, 3, 4, 5", and will generate five output data files (16T). To
16 further illustrate, "1-1000" will generate one thousand sets of output data, differing according to the
17 user's modification instructions one thousand times. In addition to simple ranging, the present
18 invention allows the user to provide step increments as well. For example, the user value "0-4; 2"
19 would be interpreted to be equivalent to entering "0, 2, 4" since the number following the semi-colon
20 describes the size of the step increment. Step increments may also be sub-integer in value, where "0-
21 4;.5" would be interpreted as entering "0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4".

22

1 [0053] In another embodiment, data formatting is also provided within the FIVS process. For
2 example, the user value “5-7: 000” is interpreted to be equivalent to entering user values of “005,
3 006, 007” where the characters following the colon describe the character filled format of the output
4 field value. Additionally, other data may be prepended or appended to the ranged values in an FIVS
5 command. For example, the user value “FIRST_”1-3”_LAST will output FIRST_1_LAST,
6 FIRST_2_LAST and FIRST_3_LAST. In one embodiment of the present invention, a special type
7 of FIVS value is provided for use in management of mandatory unique identifiers for each set of
8 output data (16), hereinafter referred to as the Trans-Session Unique Naming System (T-SUNS).

9

10 [0054] T-SUNS allows the user to designate which field(s) (24) will be modified for each set of
11 output data (16). T-SUNS is an improvement upon FIVS. Specifically, FIVS is capable of
12 understanding single fields (24) and the set or range of values (54) that the user intends for FIVS to
13 place in modified fields (24A) during output data (16) generation. However, output data (16) is
14 typically used for processing within an external system. This means that the external system may
15 require each document to be equipped with a unique identification number.

16

17 [0055] T-SUNS accommodates this requirement through a triggering command (not shown) entered
18 by the user during the FIVS process for a predetermined data field or fields. In one embodiment, the
19 triggering command instructs the system to insert a new value for the field (24) or fields having an
20 attached triggering command. This new value may then be used as a unique identifier to not only the
21 modified data fields but also for the analyzed data from which the output data (16) is created. The
22 present invention uses structural information to accomplish this. By maintaining information
23 regarding the structural characteristics of the field(s) (24) to which a unique identification has been

1 assigned, the present invention is capable of generating the unique number when required for any
2 and all output data (16) containing fields (24) with an attached triggering command.

3

4 [0056] In one embodiment, the unique identification number consists of a preamble, amble and
5 postamble. The present invention allows the user to edit the identification number as desired. In one
6 embodiment, the user may edit the preamble and postambles as desired as well as set the starting
7 value and format for the amble portion. For example, a typical use for unique identification numbers
8 is for purchase orders. This type of document typically uses numeric counters surrounded by
9 alphabetic characters and formatted with leading 0's. To illustrate, a purchase order number of
10 XYZ0001PO may be managed within the T-SUNS process as XYZ for the preamble, 1:0000 as the
11 amble and PO as the postamble. Starting with these values, T-SUNS would return XYZ0001PO the
12 first time, XYZ0002PO the second time, and so on. Since the structural characteristics are used for
13 the preamble, amble and postamble, subsequent output data (16), even if produced subsequently, are
14 capable of using the next available increment of the identification number as long as the original
15 output data's (16) structural characteristics still apply.

16

17 [0057] In addition to the above, the present invention is capable of automatically modifying output
18 data (16) through the use of the Field Instance Bounds Generation (FIBG) process. Unlike FIVS
19 where the user enters all data modification instructions (80), FIBG uses the structural characteristics
20 of the output data (16) to determine the boundaries of valid data for selected fields (24A). Once this
21 is accomplished, the FIBG process of the present invention "pushes" the boundaries. For example,
22 FIBG is capable of producing output data (16) based on the following boundaries.

23

1 [0058] A first boundary employed by one embodiment of the present invention is referred to as
2 minimum minus. Minimum minus tests use the structural characteristics of the output data (16) to
3 determine the minimum value for a given field (24). In one embodiment, output data (16) includes
4 a first value that meets the predetermined minimum value as well as a second value that decrements
5 the first value by a factor of one. Thus, the present invention may presume that output data (16)
6 having the first value which meets the predetermined minimum value will pass subsequent
7 processing (e.g., a positive test) while the output data (16) having the second value that does not
8 meet the predetermined minimum value will not pass subsequent processing (e.g., a negative test).
9 In one embodiment, the presumption of success or failure is indicated in the naming convention for
10 each set of output data (16) to provide the user with simplified review and identification of each set
11 of output data.

12
13 [0059] A second boundary employed by one embodiment of the present invention is referred to as
14 maximum plus. Maximum plus uses the structural characteristics to determine the maximum value
15 in order to output one data set for each field (24) that meets the maximum plus value, as well as one
16 data set for each field that exceed the maximum plus value. In one embodiment, the presumption of
17 success or failure is indicated in the naming convention for each set of output data (16) to provide
18 the user with simplified review and identification of each set of output data.

19
20 [0060] A third boundary employed by one embodiment of the present invention is referred to as
21 blank field. Using this boundary, blank spaces, instead of alphanumeric characters, are used for the
22 predetermined field. A fourth boundary employed by one embodiment of the present invention is
23 referred to as field type. Field type uses the structural characteristics of the field type at issue so that

1 each field may be varied. For example, typical field types include alpha, numeric and alphanumeric.
2 Fields marked for this type of output data generation will produce three data sets, one with all alpha
3 characters, one with only numeric characters, and one with mixed alphanumeric characters. In one
4 embodiment, the presumption of success or failure is based on the field type at issue.

5

6 [0061] A fifth boundary employed by one embodiment of the present invention is referred to as
7 decimal count. Decimal count is utilized primarily in conjunction with fields (24) that indicate a
8 numeric type or have a predefined decimal format. FIBG output data (16) having these fields will
9 increment and decrement the decimal position for the field's data. The presumption of success or
10 failure is based on the existing decimal format for the field, with any matching format assuming
11 success, and any deviation from the standard format assuming failure.

12

13 [0062] Although the invention has been described with reference to specific embodiments, this
14 description is not meant to be construed in a limited sense. Various modifications of the disclosed
15 embodiments, as well as alternative embodiments of the inventions will become apparent to persons
16 skilled in the art upon the reference to the description of the invention. It is, therefore, contemplated
17 that the appended claims will cover such modifications that fall within the scope of the invention.

18